Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Computational Physics 196 (2004) 208–232

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

Numerical simulation of three dimensional pyramid quantum dot

Tsung-Min Hwang [a], Wen-Wei Lin [b], Wei-Cheng Wang [b], Weichung Wang [c,*]

[a] *Department of Mathematics, National Taiwan Normal University, Taipei 116, Taiwan*
[b] *Department of Mathematics, National Tsing Hua University, Hsinchu 300, Taiwan*
[c] *Department of Applied Mathematics, National University of Kaohsiung, Kaohsiung 811, Taiwan*

Received 30 May 2003; received in revised form 27 October 2003; accepted 28 October 2003

Abstract

We present a simple and efficient numerical method for the simulation of the three-dimensional pyramid quantum dot heterostructure. The pyramid-shaped quantum dot is placed in a computational box with uniform mesh in Cartesian coordinates. The corresponding Schrödinger equation is discretized using the finite volume method and the interface conditions are incorporated into the discretization scheme without explicitly enforcing them. The resulting matrix eigenvalue problem is then solved using a Jacobi–Davidson based method. Both linear and non-linear eigenvalue problems are simulated. The scheme is 2nd order accurate and converges extremely fast. The superior performance is a combined effect of the uniform spacing of the grids and the nice structure of the resulting matrices. We have successfully simulated a variety of test problems, including a quintic polynomial eigenvalue problem with more than 32 million variables.

Keywords: Finite volume method; Heterostructure; Large scale polynomial eigenvalue problem; Semiconductor pyramid quantum dot; Schrödinger equation

1. Introduction

The purpose of this paper is to compute all the relevant energy states (eigenvalues) and the corresponding wave functions (eigenvectors) of a three-dimension (3D) semiconductor pyramidal quantum dot (QD) for electrons in the conduction band. As shown in Fig. 1, the dot is embedded in a cuboid matrix of different material. A typical example is an InAs pyramid QD embedded in a cuboid GaAs matrix.

[*] Corresponding author. Tel.: +886-7-5919521; fax: +886-7-5919344.
E-mail addresses: min@math.ntnu.edu.tw (T.-M. Hwang), wwlin@am.nthu.edu.tw (W.-W. Lin), wangwc@math.nthu.edu.tw (W.-C. Wang), wwang@nuk.edu.tw (W. Wang).

0021-9991/$ - see front matter © 2003 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2003.10.026

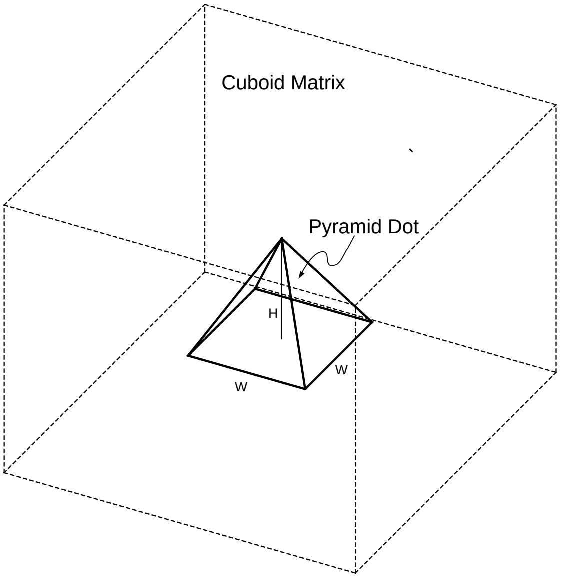

Fig. 1. Structure schema of a pyramid quantum dot.

The pyramid-shaped QDs are standard product of modern semiconductor manufacturing procedures. These nano-scale QDs have been of great interest over the past few years and have stimulated numerous research activities concerning their physical properties [9,18,22,29] and applications [6,15,16,21,28].

The governing equation for this problem is the Schrödinger equation

$$-\nabla\cdot\left(\frac{\hbar^2}{2m(\mathbf{r},\lambda)}\nabla u\right)+V(\mathbf{r})u=\lambda u, \tag{1}$$

where $\hbar$ is the reduced Plank constant, λ is the unknown eigenvalue and $u(\mathbf{r})$ the corresponding eigenfunction. Here the effective electron mass $m(\mathbf{r},\lambda)$ and the confinement potential $V(\mathbf{r})$ are discontinuous across the heterojunction.

The dependence of $m(\mathbf{r},\lambda)$ on λ can be derived from the eight-band $\mathbf{k}\cdot\mathbf{p}$ analysis and the effective mass theory. This is the simplest possible model taking into account the spin–orbit split-off coupling.

First of all, starting from the Hamiltonian of the electron with a nearly periodic potential $V_{\mathrm{p}}(\mathbf{r})$ which varies on a small scale (the lattice constant), we apply the Löwdin perturbation expansion up to second order to get an 8×8 Hamiltonian $\mathscr{H}_e$ that acts on the envelope functions for electrons in the lowest conduction band and the top valence band. The corresponding potential term $V(\mathbf{r})$ is the local average of $V_{\mathrm{p}}(\mathbf{r})$. Both $V(\mathbf{r})$ and the envelope functions vary on a scale much larger than the lattice constant.

Secondly, instead of solving the eigenvalue problem for the Hamiltonian $\mathscr{H}_e$, the effective mass theory proposes to project the 8×8 Hamiltonian onto the conduction band and results in a single Hamiltonian eigenvalue problem (1) with the effective mass $m(\lambda)$ given by the secular equation

$$\det(\mathscr{H}_e-\lambda)=0. \tag{2}$$

With properly chosen basis functions, the effective Hamiltonian $\mathscr{H}_e$ can be block-diagonalized and the solution to the secular equation (2) gives the effective mass as a rational function of the energy:

$$m(\mathbf{r},\lambda)=\begin{cases} m_1(\lambda) & \text{in material 1 (the dot),}\\ m_2(\lambda) & \text{in material 2 (the matrix),}\end{cases}\qquad V(\mathbf{r})=\begin{cases} V_1 & \text{in the dot,}\\ V_2 & \text{in the matrix,}\end{cases} \tag{3}$$

where

$$\frac{1}{m_\ell(\lambda)} = \frac{P_\ell^2}{\hbar^2}\left(\frac{2}{\lambda + g_\ell - V_\ell} + \frac{1}{\lambda + g_\ell - V_\ell + \delta_\ell}\right), \quad \ell = 1, 2. \tag{4}$$

Here P_ℓ, g_ℓ and δ_ℓ are the momentum matrix element, the conduction and spin–orbit split-off band gaps for the dot ($\ell = 1$) and the matrix ($\ell = 2$), respectively.

In the small wave number limit, $m(\lambda)$ can be further approximated by the effective mass at the Brillouin zone center. Therefore $m(\mathbf{r}, \lambda)$ and $V(\mathbf{r})$ are both piecewise constant functions of $\mathbf{r}$:

$$m(\mathbf{r}, \lambda) = \begin{cases} m_1 & \text{in the dot,} \\ m_2 & \text{in the matrix,} \end{cases} \quad V(\mathbf{r}) = \begin{cases} V_1 & \text{in the dot,} \\ V_2 & \text{in the matrix.} \end{cases} \tag{5}$$

The approximation (5) gives a quadratic dispersion relation between the energy and the wave number. Thus (5) is sometimes referred to as the parabolic approximation and (4) the non-parabolic approximation. A detail introduction of the $\mathbf{k} \cdot \mathbf{p}$ theory and the Kane model mentioned above can be found, for example, in [3,8].

An improvement over the non-parabolic approximation (4) is to take into account the effect of strain. The strain tensor is obtained using linear elastic theory via minimization of the free energy for the structure [33]. The computed strain is then added to the effective Hamiltonian $\mathscr{H}_e$ following the Bir–Pikus theory [4]. The presence of the strain Hamiltonian changes the dispersion relation and the band gaps of the materials. Since the strain is anisotropic and position dependent, the resulting effective mass is a tensor and the corresponding Schrödinger equation becomes

$$\left(-\frac{\hbar^2}{2}\sum_{\alpha,\beta=x,y,z} \partial_\alpha \frac{1}{m_{\alpha\beta}(\mathbf{r}, \lambda)} \partial_\beta + V(\mathbf{r})\right) u = \lambda u. \tag{6}$$

A detailed derivation starting from the six-band Kane model with strain (neglecting the spin–orbit split-off coupling) can be found in [14]. Following the notations in [12,13], the perpendicular and in-plane components of the (diagonal) effective mass tensor are given by

$$\begin{aligned} \frac{1}{m_{xx}(\mathbf{r}, \lambda)} &= \frac{1}{m_{yy}(\mathbf{r}, \lambda)} = \frac{E_\mathrm{g}}{m^*}\left(\frac{0.25}{\lambda - V_{lh}(\mathbf{r})} + \frac{0.75}{\lambda - V_{hh}(\mathbf{r})}\right) \\ \frac{1}{m_{zz}(\mathbf{r}, \lambda)} &= \frac{E_\mathrm{g}}{m^*}\left(\frac{1}{\lambda - V_{lh}(\mathbf{r})}\right), \end{aligned} \tag{7}$$

where m^* and E_g denote the bulk effective mass and band gap of InAs, and $V_{hh}(\mathbf{r})$ and $V_{lh}(\mathbf{r})$ the position dependent heavy hole and light-hole bands of the structure.

The potential term $V(\mathbf{r})$ in (6) include the contributions from externally applied voltage, the conduction-band offset, the conduction-band strain potential, and the piezoelectric potential. The contribution from the strain tensor is also included in the band edge energies $V_c(\mathbf{r})$, $V_{hh}(\mathbf{r})$ and $V_{lh}(\mathbf{r})$ [12,14].

A constant effective mass model derived from the six-band model with strain (excluding the conduction band) can be found in [8].

If in addition, the diffusion of Indium is taken into account, the material constants for $\mathrm{In}_x\mathrm{Ga}_{1-x}\mathrm{As}$ can be taken as, for example, the interpolation of those of InAs and GaAs [8,12] and the effective mass tensor remains a rational function of the form (7). An explicit form of the Indium mole fraction distribution $x(\mathbf{r})$ was proposed in [12].

In summary, the effective mass (3) and (7) can be expressed as a rational function in λ of the form

$$\frac{1}{m(\mathbf{r},\lambda)}=\sum_{i}^{n}\frac{c_i(\mathbf{r})}{\lambda-W_i(\mathbf{r})},\tag{8}$$

where $c_i(\mathbf{r})$, $W_i(\mathbf{r})$ and the potential term $V(\mathbf{r})$ are position dependent functions which are discontinuous across the heterojunction. This motivates us to propose an efficient numerical method for solving the generalized eigenvalue problem (1). Our method applies to the general case (8). For simplicity of presentation, we will only consider (4) and (5) throughout this paper.

Due to the complicated nature of the Schrödinger equation (1), the analytic solutions can only be obtained for 1D (quantum wells) and 2D (quantum wires) models with symmetry. Relatively few research reports are focused on real 3D QDs. In such cases, neither analytical techniques nor asymptotic analysis gives useful information and numerical simulation becomes a very important access to this problem.

While numerical simulations play an important role [32,36,44], most existing methods are designed for 1D and 2D problems with symmetry where the matrix size of the eigenvalue problem is much smaller. Besides, very few results can be applied by current computational methods for real 3D QDs [17, Section 11.6].

There are a few recent progress on various models of genuine 3D QD simulations. Pryor [33] studied the electronic structure of pyramidal shaped QDs. An eight-band strain-dependent $\mathbf{k}\cdot\mathbf{p}$ Hamiltonian was discretized by finite differences and then solved by the Lanczos algorithm. El-Moghraby et al. [10] used a finite difference technique to solve the Schrödinger equation of a pyramid QD with constant effective mass model. Instead of solving the eigenvalue problems, the energies (eigenvalues) were scanned over a range and each of the chosen energies are substituted into the discretized equation to form a linear system and determine if the corresponding matrix is singular. All of these systems are examined to obtain the candidates of the wave function (eigenvector).

In this paper, we propose to solve the linear and non-linear eigenvalue problems directly using a finite volume discretization and a Jacobi–Davidson based eigensolver.

Since the 3D problem results in a huge matrix eigenvalue problem, the discretization of the Schrödinger equation much affects the performance of the overall scheme. From our experience in the simulation of cylindrical QDs, first order discretization gives relatively low accuracy, especially for non-linear eigenvalue problems [26,43]. As a result, local mesh refinement near the heterojunction is necessary in order to achieve reasonable accuracy. However, local mesh refinement also destroys the structure of the matrices. The diagonal entries corresponding to the fine meshes are much larger than others. The variations among the diagonal entries made efficient preconditioning quite difficult. This is even more challenging for large scale non-linear problems.

Fortunately, thanks to the geometric structure of the pyramid, the QD can be embedded in a Cartesian coordinate with uniform mesh. We will show that, with a carefully chosen grids, the Schrödinger equation can be discretized using a finite volume method that automatically builds in the interface condition and maintain global second order accuracy.

The discretized Schrödinger equation then leads to the matrix polynomial eigenvalue problem

$$\mathbf{A}(\lambda)\mathbf{F}\equiv\left(\sum_{i=0}^{\tau}\lambda^i A_i\right)\mathbf{F}=0,\tag{9}$$

where $\lambda\in\mathbb{C}$, $\mathbf{F}\in\mathbb{C}^{\mathscr{N}}$, τ is the degree of the matrix polynomial, $A_i\in\mathbb{R}^{\mathscr{N}\times\mathscr{N}}$, and $\mathscr{N}$ is the total number of unknowns. The matrix polynomial eigenvalue problem (9) is a linear ($\tau=1$) eigenvalue problem for the constant effective mass model (5). It becomes a quintic or higher order polynomial eigenvalue problem ($\tau\geqslant 5$) provided the models (4) or (8) is used. The main computational challenge is that only several smallest positive eigenvalues of a very large-sized polynomial eigenvalue problem are relevant.

Traditional linear eigenvalue problem solvers, such as rational Krylov subspace methods [24,34,35], a linear system is solved approximately in each of the iterative steps. This process usually leads to intractable computational cost and sometimes inaccurate eigenpairs for large eigenvalue problems that typically arises in 3D problems. In contrast, recently developed Jacobi–Davidson methods [30,37–39] suggest a promising approach for linear and quadratic eigenvalue problems of very large size. Instead of inverting a large sparse matrix, the main source of computational cost for this approach is the matrix-vector multiplication. Only a much smaller (usually tens of unknowns) eigenvalue problem is actually solved. Inspired by the results reported in [30,37–39], we propose a modified version of the Jacobi–Davidson methods with locking and restarting deflation technique to solve the desired eigenpairs of (9).

Overall, we find the resulting spatial discretization to be second order accurate and the eigensolver converges extremely fast. The uniform placement of the grids not only reduces the programming labor, it is also crucial in accelerating the convergence of the Jacobi–Davidson step. From our simulation using $64 \times 64 \times 48$ grids, the computed eigenvalues have only 0.2% to 0.2% relative error from its limiting value ($h \to 0$). On the other hand, since the matrices are sparse, we are able to increase the matrix size to the machine memory limit (about 8 gigabytes) that corresponds to more than 32 million unknowns, the resulting eigenvalues and eigenvectors converges within 74–220 iterations, or 4800–8200 s on a decent workstation.

The rest of the paper is organized as follows. We first derive the finite volume scheme over a 2D triangular and a 3D pyramid domain in Sections 2.1 and 3. The Jacobi–Davidson type method for solving the corresponding eigenvalue problems is then described in Section 4. Numerical examples are reported and analyzed in Section 5 to explore the performance of the scheme. Finally we close the paper with a few concluding remarks in Section 6.

2. The discretization of the Schrödinger equation

The Schrödinger equations (1) and (10) with discontinuous effective mass is also known as the elliptic interface problem. Associated with the discontinuity in m, we have the following interface condition (usually refereed to as the Ben Daniel–Duke condition [3])

$$\left(\frac{1}{m(\mathbf{r},\lambda)}\frac{\partial u}{\partial n}\right)\bigg|_{\partial D_-} = \left(\frac{1}{m(\mathbf{r},\lambda)}\frac{\partial u}{\partial n}\right)\bigg|_{\partial D_+}, \tag{10}$$

where D is the domain of the pyramid dot and n is the outward normal.

The interface problem, in general, may be solved by other second order methods such as the immersed interface method (IIM) [25], body fitting finite element method [5,7] or the immersed finite element method [11]. However, most of these methods are designed for piecewise constant coefficient problems and become quite complicated, sometimes inaccessible, for polynomial eigenvalue problems. There is also a class of finite difference schemes that replace the discontinuous coefficient by a smooth coefficient using an averaging process and then discretize the modified problem with standard centered difference. These scheme are known to have only first order accuracy and are thus not recommended for this problem.

Due to the special structure of the pyramid, we can embed the QD in a Cartesian coordinate with uniform mesh. We will show that the corresponding finite volume discretization results in a monotone operator with $O(h)$ local truncation error on the interface and $O(h^2)$ elsewhere. Therefore global second order accuracy is expected. This is indeed verified in our numerical examples. See Section 5.

For simplicity of presentation, we will give detail derivation of the scheme over 2D triangular domain and generalize the result to 3D pyramid domain.

2.1. 2D finite volume scheme for a triangular domain

Consider a 2D triangular QD with height H and base width W placed in rectangular domain. In practice, we are only interested in the confined eigenfunctions of (1) which decay exponentially outside the QD. We therefore choose a $2W \times 3H$ rectangle and impose the homogeneous Dirichlet boundary conditions $u = 0$ on the outer boundary of the rectangle for simplicity. We fix the ratio of the mesh sizes $\Delta x/\Delta y = W/2H$ as mesh refines, see Fig. 2. For convenience, the nodes in the exterior and interior of the triangle are labeled by 0 and 1, respectively. Those on the left, right, and bottom of the triangle are labeled by 2, 3, and 4; the top, left, and right corners are labeled by 5, 6, and 7, respectively.

For simplicity, we use the notation

$$\alpha^+ \equiv \frac{\hbar^2}{2m_2} \quad \text{and} \quad \alpha^- \equiv \frac{\hbar^2}{2m_1} \tag{11}$$

to rewrite the Schrödinger equation (1) and the interface condition (10) as

$$-\nabla \cdot (\alpha \nabla u) + Vu = \lambda u, \tag{12}$$

where

$$\alpha = \begin{cases} \alpha^- & \text{inside,} \\ \alpha^+ & \text{outside,} \end{cases} \qquad V = \begin{cases} V^- = V_1 & \text{inside,} \\ V^+ = V_2 & \text{outside} \end{cases}$$

and

$$\alpha^- \frac{\partial u}{\partial n}\bigg|_{\partial D_-} = \alpha^+ \frac{\partial u}{\partial n}\bigg|_{\partial D_+}, \tag{13}$$

respectively.

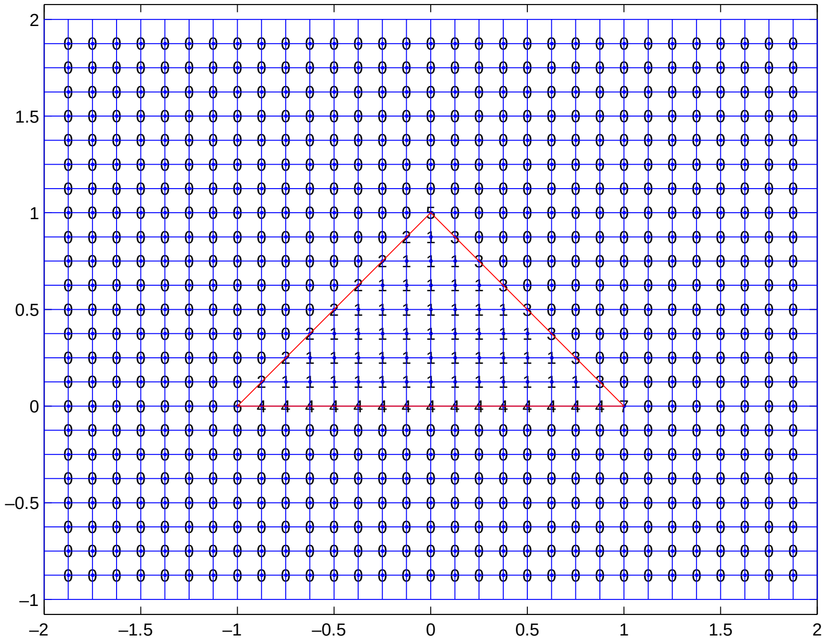

Fig. 2. Schema of the uniform discretization scheme of a 2D domain.

Consider the standard centered difference method for points in the interior and the exterior

$$-\alpha\left(\frac{u_{i+1,j}-2u_{i,j}+u_{i-1,j}}{(\Delta x)^2}+\frac{u_{i,j+1}-2u_{i,j}+u_{i,j-1}}{(\Delta y)^2}\right)+Vu_{i,j}=\lambda u_{i,j}. \tag{14}$$

Since all five points are on the same side of the heterojunction, α and V are therefore a constants among all five points and the local truncation error for (14) is $O(h^2)$ from standard analysis.

The standard finite difference scheme (14) can also be rearranged as

$$-\alpha\left(\frac{u_{i+1,j}-u_{i,j}}{\Delta x}\Delta y+\frac{u_{i-1,j}-u_{i,j}}{\Delta x}\Delta y+\frac{u_{i,j+1}-u_{i,j}}{\Delta y}\Delta x+\frac{u_{i,j-1}-u_{i,j}}{\Delta y}\Delta x\right)=\Delta x\Delta y(\lambda-V)u_{i,j}, \tag{15}$$

which is a 2nd order approximation of the integral form of the Schrödinger equation (1) over the control volume Ω of size $\Delta x\times\Delta y$

$$-\int_{\partial\Omega}\alpha\frac{\partial u}{\partial n}=\int\int_{\Omega}(\lambda-V)u \tag{16}$$

as shown in (e) of Fig. 3.

This finite volume interpretation for (14) is the basis for the discretization of nodes on the heterojunction (labeled 2 through 7).

For example, consider the mesh points located at the left hypotenuse and the integral equation (16) over the control volume Ω as shown in (d) of Fig. 3. We denote the four portions of $\partial\Omega$ as $\partial\Omega_E$, $\partial\Omega_N$, $\partial\Omega_W$, and $\partial\Omega_S$, respectively.

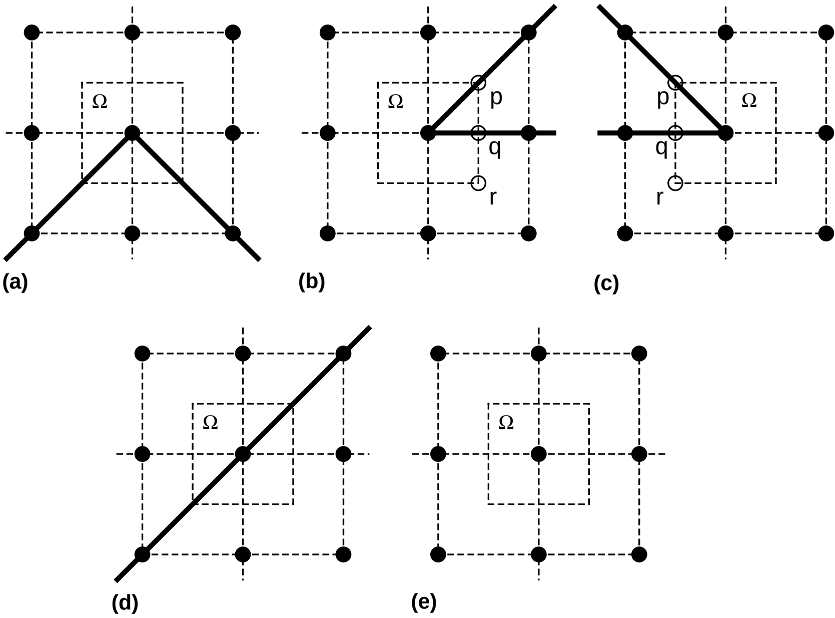

Fig. 3. Discretization schema of the (a) top, (b) left, (c) right corners and (d) left hypotenuse. The solid lines represents the heterojunctions. The solid points are the mesh points.

The line integral over, say, $\partial\Omega_{\rm E}$ can be approximated by

$$\int_{\partial\Omega_{\rm E}} \alpha u_n \, {\rm d}s = \int_{\partial\Omega_{\rm E}} \alpha^- u_x \, {\rm d}s = \alpha^- u'_{i+\frac{1}{2},j}\Delta y + {\rm O}(\Delta y^3) = \alpha^- \frac{u_{i+1,j} - u_{i,j}}{\Delta x}\Delta y + {\rm O}(h^3),$$

where $h = \max(\Delta x, \Delta y)$ and we can therefore approximate the left hand side of Eq. (16) by

$$\begin{aligned} -\int_{\partial\Omega} \alpha\frac{\partial u}{\partial n} &= -\int_{\partial\Omega_{\rm E}} \alpha\frac{\partial u}{\partial n} - \int_{\partial\Omega_{\rm N}} \alpha\frac{\partial u}{\partial n} - \int_{\partial\Omega_{\rm W}} \alpha\frac{\partial u}{\partial n} - \int_{\partial\Omega_{\rm S}} \alpha\frac{\partial u}{\partial n} \\ &= -\alpha^- \frac{u_{i+1,j} - u_{i,j}}{\Delta x}\Delta y - \alpha^+ \frac{u_{i,j+1} - u_{i,j}}{\Delta y}\Delta x - \alpha^+ \frac{u_{i-1,j} - u_{i,j}}{\Delta x}\Delta y - \alpha^- \frac{u_{i,j-1} - u_{i,j}}{\Delta y}\Delta x + {\rm O}(h^3). \end{aligned} \tag{17}$$

On the other hand, the right hand side of Eq. (16) can be approximated by

$$\int\int_{\Omega} (\lambda - V)u = \Delta x \Delta y\Big(- \Big(\frac{1}{2}V^- + \frac{1}{2}V^+\Big)u_{i,j} + \lambda u_{i,j} + {\rm O}(h)\Big). \tag{18}$$

It follows that the finite volume discretization on the left hypotenuse is given by

$$\begin{aligned} &-\frac{1}{(\Delta x)^2}\big(\alpha^+ u_{i-1,j} - (\alpha^+ + \alpha^-)u_{i,j} + \alpha^- u_{i+1,j}\big) - \frac{1}{(\Delta y)^2}\big(\alpha^- u_{i,j-1} - (\alpha^+ + \alpha^-)u_{i,j} + \alpha^+ u_{i,j+1}\big) \\ &\quad + \Big(\frac{1}{2}V^- + \frac{1}{2}V^+\Big)u_{i,j} = \lambda u_{i,j} + {\rm O}(h). \end{aligned} \tag{19}$$

Similarly, we can derive the finite volume discretization on the right hypotenuse

$$\begin{aligned} &-\frac{1}{(\Delta x)^2}\big(\alpha^- u_{i-1,j} - (\alpha^+ + \alpha^-)u_{i,j} + \alpha^+ u_{i+1,j}\big) - \frac{1}{(\Delta y)^2}\big(\alpha^- u_{i,j-1} - (\alpha^+ + \alpha^-)u_{i,j} + \alpha^+ u_{i,j+1}\big) \\ &\quad + \Big(\frac{1}{2}V^- + \frac{1}{2}V^+\Big)u_{i,j} = \lambda u_{i,j} + {\rm O}(h) \end{aligned} \tag{20}$$

and on the top

$$\begin{aligned} &-\frac{1}{(\Delta x)^2}\big(\alpha^+ u_{i-1,j} - 2\alpha^+ u_{i,j} + \alpha^+ u_{i+1,j}\big) - \frac{1}{(\Delta y)^2}\big(\alpha^- u_{i,j-1} - (\alpha^+ + \alpha^-)u_{i,j} + \alpha^+ u_{i,j+1}\big) \\ &\quad + \Big(\frac{1}{4}V^- + \frac{3}{4}V^+\Big)u_{i,j} = \lambda u_{i,j} + {\rm O}(h), \end{aligned} \tag{21}$$

respectively.

The derivation for the finite volume approximation on the left and the right corners is similar. We divide $\partial\Omega_{\rm E}$ into two segments $\overline{rq}$ and $\overline{qp}$ (Fig. 3(b)) and then use the local expansions

$$\begin{aligned} u_x^-(p) &= u_x^-(q) + \frac{\Delta y}{2}u_{yx}^-(q) + {\rm O}(h^2), \\ u_x(r) &= u_x^+(q) - \frac{\Delta y}{2}u_{yx}^+(q) + {\rm O}(h^2) \end{aligned} \tag{22}$$

to compute the east line integral. Here the superscript denotes the limiting value from inside (−) and outside (+), respectively.

From (22), we have

$$\begin{aligned}
\int_{\partial\Omega_{\mathrm{E}}} \alpha \frac{\partial u}{\partial n}\,\mathrm{d}\ell_y &= \int_{\partial\Omega_{\mathrm{E}}} \alpha u_x\,\mathrm{d}\ell_y = \int_q^p \alpha^- u_x\,\mathrm{d}\ell_y + \int_r^q \alpha^+ u_x\,\mathrm{d}\ell_y \\
&= \alpha^-\left(\frac{u_x(p)+u_x(q)}{2}\right)\frac{\Delta y}{2} + \alpha^+\left(\frac{u_x(q)+u_x(r)}{2}\right)\frac{\Delta y}{2} + \mathrm{O}(h^3) \quad \text{(trapezoidal rule)} \\
&= \alpha^-\left(u_x(q) + \frac{u_x^-(p)-u_x^-(q)}{2}\right)\frac{\Delta y}{2} + \alpha^+\left(u_x^+(q) + \frac{u_x(r)-u_x^+(q)}{2}\right)\frac{\Delta y}{2} + \mathrm{O}(h^3) \\
&= (\alpha^- + \alpha^+)\frac{\Delta y}{2}u_x(q) + \left(\frac{\Delta y}{2}\right)^2\left(\frac{1}{2}\right)\left(\alpha^- u_{yx}^-(q) - \alpha^+ u_{yx}^+(q)\right) + \mathrm{O}(h^3) \\
&= \frac{(\alpha^- + \alpha^+)\Delta y}{2}u_x(q) + \mathrm{O}(h^3) \\
&= \frac{(\alpha^- + \alpha^+)}{2}\,\frac{u_{i+1,j} - u_{i,j}}{\Delta x}\Delta y + \mathrm{O}(h^3),
\end{aligned}$$

where we have used the formula

$$\alpha^+ u_{yx}^+(q) = \alpha^- u_{yx}^-(q),$$

which is a direct consequence of the interface condition

$$\alpha^+ u_y^+(q) = \alpha^- u_y^-(q).$$

We can now approximate the left hand side of Eq. (16) at the left corner as

$$\begin{aligned}
-\int_{\partial\Omega} \alpha\frac{\partial u}{\partial n} &= -\int_{\partial\Omega_{\mathrm{N}}} \alpha\frac{\partial u}{\partial n} - \int_{\partial\Omega_{\mathrm{W}}} \alpha\frac{\partial u}{\partial n} - \int_{\partial\Omega_{\mathrm{S}}} \alpha\frac{\partial u}{\partial n} - \int_{\partial\Omega_{\mathrm{E}}} \alpha\frac{\partial u}{\partial n} \\
&= -\alpha^+\frac{u_{i,j+1} - u_{i,j}}{\Delta y}\Delta x - \alpha^+\frac{u_{i-1,j} - u_{i,j}}{\Delta x}\Delta y - \alpha^+\frac{u_{i,j-1} - u_{i,j}}{\Delta y}\Delta x \\
&\quad - \frac{(\alpha^- + \alpha^+)}{2}\,\frac{u_{i+1,j} - u_{i,j}}{\Delta x}\Delta y + \mathrm{O}(h^3).
\end{aligned} \tag{23}$$

Similarly, the right hand side of the equation is approximated by

$$\int\int_{\Omega} (\lambda - V)u = \Delta x\,\Delta y\left(\lambda u_{i,j} - \left(\frac{1}{8}V^- + \frac{7}{8}V^+\right)u_{i,j} + \mathrm{O}(h)\right) \tag{24}$$

and we conclude with the finite volume discretization at the left corner as

$$\begin{aligned}
&-\frac{1}{(\Delta x)^2}\left(\alpha^+ u_{i-1,j} - \left(\alpha^+ + \frac{\alpha^+ + \alpha^-}{2}\right)u_{i,j} + \frac{\alpha^+ + \alpha^-}{2}u_{i+1,j}\right) - \frac{1}{(\Delta y)^2}\left(\alpha^+ u_{i,j-1} - 2\alpha^+ u_{i,j} + \alpha^+ u_{i,j+1}\right) \\
&\quad + \left(\frac{1}{8}V^- + \frac{7}{8}V^+\right)u_{i,j} = \lambda u_{i,j} + \mathrm{O}(h).
\end{aligned} \tag{25}$$

The derivation for the discretization at the right corner and the bottom points are quite similar. The results are given by (the right corner)

$$
-\frac{1}{(\Delta x)^2}\left(\frac{\alpha^- + \alpha^+}{2}u_{i-1,j} - \left(\frac{\alpha^+ + \alpha^-}{2} + \alpha^+\right)u_{i,j} + \alpha^+ u_{i+1,j}\right) - \frac{1}{(\Delta y)^2}\left(\alpha^+ u_{i,j-1} - 2\alpha^+ u_{i,j} + \alpha^+ u_{i,j+1}\right)
+ \left(\frac{1}{8}V^- + \frac{7}{8}V^+\right)u_{i,j} = \lambda u_{i,j} + \mathrm{O}(h) \tag{26}
$$

and (the bottom)

$$
-\frac{1}{(\Delta x)^2}\frac{\alpha^- + \alpha^+}{2}\left(u_{i-1,j} - 2u_{i,j} + u_{i+1,j}\right) - \frac{1}{(\Delta y)^2}\left(\alpha^+ u_{i,j-1} - (\alpha^+ + \alpha^-)u_{i,j} + \alpha^- u_{i,j+1}\right)
+ \left(\frac{1}{2}V^- + \frac{1}{2}V^+\right)u_{i,j} = \lambda u_{i,j} + \mathrm{O}(h), \tag{27}
$$

respectively.

Remarks.

1. We can summarize the discretization (14), (19)–(21) and (25)–(27) as

$$
-\nabla_h \cdot (\bar{\alpha}\nabla_h u) + \bar{\bar{V}}u = \lambda u, \tag{28}
$$

where $\bar{\alpha}$ and $\bar{\bar{V}}$ denote line average of α and area average of V over the control cell, respectively.

2. The finite volume scheme presented here can be derived as a finite difference scheme which incorporates the jump condition into the discretization without explicitly enforcing them. The finite difference interpretation of the scheme allows us the to extend the current scheme into higher order finite difference scheme. The detail derivation is given in Appendix A.

3. 3D scheme for the pyramid quantum dot

The finite volume scheme over the 3D domain can be easily extended from the 2D scheme presented in Section 2, we omit the detail derivation and only summarize them as

$$
-\nabla_h \cdot (\bar{\bar{\alpha}}\nabla_h u) + \bar{\bar{\bar{V}}}u = \lambda u, \tag{29}
$$

where $\bar{\bar{\alpha}}$ and $\bar{\bar{\bar{V}}}$ denote the surface averages of α and the volume average of V over the controlled volume element, respectively.

We list the detailed formulas for representative points on the interior/exterior, surfaces, edges and corners as below:

- Points in the exterior of the pyramid:

$$
\frac{1}{(\Delta x)^2}\left\{\alpha^+ u_{i-1,j,k} - 2\alpha^+ u_{ijk} + \alpha^+ u_{i+1,j,k}\right\} + \frac{1}{(\Delta y)^2}\left\{\alpha^+ u_{i,j-1,k} - 2\alpha^+ u_{ijk} + \alpha^+ u_{i,j+1,k}\right\}
+ \frac{1}{(\Delta z)^2}\left\{\alpha^+ u_{i,j,k-1} - 2\alpha^+ u_{ijk} + \alpha^+ u_{i,j,k+1}\right\}
= V^+ u_{ijk} - \lambda u_{ijk}. \tag{30}
$$

- Points in the interior of the pyramid:

$$\begin{aligned}&\frac{1}{(\Delta x)^2}\left\{\alpha^- u_{i-1,j,k} - 2\alpha^- u_{ijk} + \alpha^- u_{i+1,j,k}\right\} + \frac{1}{(\Delta y)^2}\left\{\alpha^- u_{i,j-1,k} - 2\alpha^- u_{ijk} + \alpha^- u_{i,j+1,k}\right\}\\ &\quad + \frac{1}{(\Delta z)^2}\left\{\alpha^- u_{i,j,k-1} - 2\alpha^- u_{ijk} + \alpha^- u_{i,j,k+1}\right\}\\ &\qquad = V^- u_{ijk} - \lambda u_{ijk}.\end{aligned} \tag{31}$$

- Points on the western surface of the pyramid:

$$\begin{aligned}&\frac{1}{(\Delta x)^2}\left\{\alpha^+ u_{i-1,j,k} - (\alpha^+ + \alpha^-)u_{ijk} + \alpha^- u_{i+1,j,k}\right\} + \frac{1}{(\Delta y)^2}\left\{\frac{\alpha^+ + \alpha^-}{2}\left(u_{i,j-1,k} - 2u_{ijk} + u_{i,j+1,k}\right)\right\}\\ &\quad + \frac{1}{(\Delta z)^2}\left\{\alpha^- u_{i,j,k-1} - (\alpha^+ + \alpha^-)u_{ijk} + \alpha^+ u_{i,j,k+1}\right\}\\ &\qquad = \frac{V^+ + V^-}{2}u_{ijk} - \lambda u_{ijk}.\end{aligned} \tag{32}$$

- Points on the southern surface of the pyramid:

$$\begin{aligned}&\frac{1}{(\Delta x)^2}\left\{\frac{\alpha^+ + \alpha^-}{2}\left(u_{i-1,j,k} - 2u_{ijk} + u_{i+1,j,k}\right)\right\} + \frac{1}{(\Delta y)^2}\left\{\alpha^+ u_{i,j-1,k} - (\alpha^+ + \alpha^-)u_{ijk} + \alpha^- u_{i,j+1,k}\right\}\\ &\quad + \frac{1}{(\Delta z)^2}\left\{\alpha^- u_{i,j,k-1} - (\alpha^+ + \alpha^-)u_{ijk} + \alpha^+ u_{i,j,k+1}\right\}\\ &\qquad = \frac{V^+ + V^-}{2}u_{ijk} - \lambda u_{ijk}.\end{aligned} \tag{33}$$

- Points on the bottom surface of the pyramid:

$$\begin{aligned}&\frac{1}{(\Delta x)^2}\left\{\frac{\alpha^+ + \alpha^-}{2}\left(u_{i-1,j,k} - 2u_{ijk} + u_{i+1,j,k}\right)\right\} + \frac{1}{(\Delta y)^2}\left\{\frac{\alpha^+ + \alpha^-}{2}\left(u_{i,j-1,k} - 2u_{ijk} + u_{i,j+1,k}\right)\right\}\\ &\quad + \frac{1}{(\Delta z)^2}\left\{\alpha^+ u_{i,j,k-1} - (\alpha^+ + \alpha^-)u_{ijk} + \alpha^- u_{i,j,k+1}\right\}\\ &\qquad = \frac{V^+ + V^-}{2}u_{ijk} - \lambda u_{ijk}.\end{aligned} \tag{34}$$

- Points on the southwestern edge of the pyramid:

$$\begin{aligned}&\frac{1}{(\Delta x)^2}\left\{\alpha^+ u_{i-1,j,k} - \left(\alpha^+ + \frac{\alpha^+ + \alpha^-}{2}\right)u_{ijk} + \frac{\alpha^+ + \alpha^-}{2}u_{i+1,j,k}\right\}\\ &\quad + \frac{1}{(\Delta y)^2}\left\{\alpha^+ u_{i,j-1,k} - \left(\alpha^+ + \frac{\alpha^+ + \alpha^-}{2}\right)u_{ijk} + \frac{\alpha^+ + \alpha^-}{2}u_{i,j+1,k}\right\}\\ &\quad + \frac{1}{(\Delta z)^2}\left\{\alpha^- u_{i,j,k-1} - (\alpha^+ + \alpha^-)u_{ijk} + \alpha^+ u_{i,j,k+1}\right\} = \frac{2V^+ + V^-}{3}u_{ijk} - \lambda u_{ijk}.\end{aligned} \tag{35}$$

- Points on the western edge at the bottom of the pyramid:

$$\frac{1}{(\Delta x)^2}\left\{\alpha^+ u_{i-1,j,k} - \left(\alpha^+ + \frac{\alpha^+ + \alpha^-}{2}\right)u_{ijk} + \frac{\alpha^+ + \alpha^-}{2}u_{i+1,j,k}\right\} + \frac{1}{(\Delta y)^2}\left\{\frac{7\alpha^+ + \alpha^-}{8}\left(u_{i,j-1,k} - 2u_{ijk} + u_{i,j+1,k}\right)\right\} + \frac{1}{(\Delta z)^2}\left\{\alpha^+ u_{i,j,k-1} - 2\alpha^+ u_{ijk} + \alpha^+ u_{i,j,k+1}\right\} = \frac{7V^+ + V^-}{8}u_{ijk} - \lambda u_{ijk}. \tag{36}$$

- Points on the southern edge at the bottom of the pyramid:

$$\frac{1}{(\Delta x)^2}\left\{\frac{7\alpha^+ + \alpha^-}{8}\left(u_{i-1,j,k} - 2u_{ijk} + u_{i+1,j,k}\right)\right\} + \frac{1}{(\Delta y)^2}\left\{\alpha^+ u_{i,j-1,k} - \left(\alpha^+ + \frac{\alpha^+ + \alpha^-}{2}\right)u_{ijk} + \frac{\alpha^+ + \alpha^-}{2}u_{i,j+1,k}\right\} + \frac{1}{(\Delta z)^2}\left\{\alpha^+ u_{i,j,k-1} - 2\alpha^+ u_{ijk} + \alpha^+ u_{i,j,k+1}\right\} = \frac{7V^+ + V^-}{8}u_{ijk} - \lambda u_{ijk}. \tag{37}$$

- The southwestern corner at the bottom of the pyramid:

$$\frac{1}{(\Delta x)^2}\left\{\alpha^+ u_{i-1,j,k} - \left(\alpha^+ + \frac{7\alpha^+ + \alpha^-}{8}\right)u_{ijk} + \frac{7\alpha^+ + \alpha^-}{8}u_{i+1,j,k}\right\} + \frac{1}{(\Delta y)^2}\left\{\alpha^+ u_{i,j-1,k} - \left(\alpha^+ + \frac{7\alpha^+ + \alpha^-}{8}\right)u_{ijk} + \frac{7\alpha^+ + \alpha^-}{8}u_{i,j+1,k}\right\} + \frac{1}{(\Delta z)^2}\left\{\alpha^+ u_{i,j,k-1} - 2\alpha^+ u_{ijk} + \alpha^+ u_{i,j,k+1}\right\} = \frac{23V^+ + V^-}{24}u_{ijk} - \lambda u_{ijk}. \tag{38}$$

- The tip of the pyramid:

$$\frac{1}{(\Delta x)^2}\left\{\alpha^+ u_{i-1,j,k} - 2\alpha^+ u_{ijk} + \alpha^+ u_{i+1,j,k}\right\} + \frac{1}{(\Delta y)^2}\left\{\alpha^+ u_{i,j-1,k} - 2\alpha^+ u_{ijk} + \alpha^+ u_{i,j+1,k}\right\} + \frac{1}{(\Delta z)^2}\left\{\alpha^- u_{i,j,k-1} - (\alpha^+ + \alpha^-)u_{ijk} + \alpha^+ u_{i,j,k+1}\right\} = \frac{5V^+ + V^-}{6}u_{ijk} - \lambda u_{ijk}. \tag{39}$$

Remark.

1. The finite volume discretization (28) or (29) can be applied, in a straightforward manner, to arrays of QDs using Cartesian coordinates. It also extends easily to 2D or 3D truncated pyramid dots [12]. For example, the discretization on the left top corner of a 2D truncated dot (point P in Fig. 4(a)) is given by

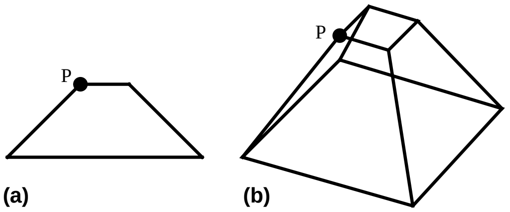

Fig. 4. Structure schemes of truncated pyramid quantum dots in 2D and 3D.

$$-\frac{1}{(\Delta x)^2}\left(\alpha^+ u_{i-1,j} - \left(\alpha^+ + \frac{\alpha^+ + \alpha^-}{2}\right)u_{i,j} + \frac{\alpha^+ + \alpha^-}{2}u_{i+1,j}\right)$$
$$-\frac{1}{(\Delta y)^2}\left(\alpha^- u_{i,j-1} - (\alpha^- + \alpha^+)u_{i,j} + \alpha^+ u_{i,j+1}\right) + \left(\frac{3}{8}V^- + \frac{5}{8}V^+\right)u_{i,j} = \lambda u_{i,j} + \mathrm{O}(h).$$

The discretization on the southwestern corner at the top of a truncated 3D pyramid (point P in Fig. 4(b)) is given by

$$\frac{1}{(\Delta x)^2}\left\{\alpha^+ u_{i-1,j,k} - \left(\alpha^+ + \frac{5\alpha^+ + 3\alpha^-}{8}\right)u_{ijk} + \frac{5\alpha^+ + 3\alpha^-}{8}u_{i+1,j,k}\right\}$$
$$+\frac{1}{(\Delta y)^2}\left\{\alpha^+ u_{i,j-1,k} - \left(\alpha^+ + \frac{5\alpha^+ + 3\alpha^-}{8}\right)u_{ijk} + \frac{5\alpha^+ + 3\alpha^-}{8}u_{i,j+1,k}\right\}$$
$$+\frac{1}{(\Delta z)^2}\left\{\alpha^- u_{i,j,k-1} - (\alpha^- + \alpha^+)u_{ijk} + \alpha^+ u_{i,j,k+1}\right\} = \frac{17V^+ + 7V^-}{24}u_{ijk} - \lambda u_{ijk}.$$

2. For QDs of general shape, e.g. lens shaped ones, a more sophisticated discretization method is needed. We have adopted the finite difference method on a body-fitting curvilinear coordinate system developed in [20]. The test results are quite encouraging and the details will be reported elsewhere.

4. Jacobi–Davidson based algorithm for eigenvalue problems

The finite volume discretization (29) results in various eigenvalue problems. For the constant coefficient case, α and V are piecewise constants, we have the linear eigenvalue problem

$$A_0\mathbf{F} = -\lambda A_1\mathbf{F}, \tag{40}$$

where A_1 is the identity matrix and A_0 is a sparse matrix with non-zero entries located in the main diagonal and six off-diagonals. Furthermore, since the matrix A_0 is symmetric and positive definite, the eigenvalues of (40) are all positive. In other words, the desired eigenvalues are the ones located in the lower end of the matrix spectrum.

For the non-parabolic model (4), $V(\mathbf{r})$ is piecewise constant and α is a rational function of the λ with piecewise constant coefficients. At each grid point, the discretization (29) results in a rational function of λ. By multiplying the common denominators of the rational functions

$$(\lambda + g_1 - V_1)(\lambda + g_1 - V_1 + \delta_1)$$

and

$$(\lambda + g_2 - V_2)(\lambda + g_2 - V_2 + \delta_2)$$

for the grids in the interior and exterior, respectively, Eqs. (30) and (31) become cubic functions of λ. For all interface grid points (32)–(39), we multiply (29) by the common denominator

$$(\lambda + g_1 - V_1)(\lambda + g_1 - V_1 + \delta_1)(\lambda + g_2 - V_2)(\lambda + g_2 - V_2 + \delta_2).$$

The result is a quintic polynomial of λ. With elementary algebraic manipulation, the discretization (29) is reduced to a quintic polynomial eigenvalue problem:

$$(\lambda^5 A_5 + \lambda^4 A_4 + \lambda^3 A_3 + \lambda^2 A_2 + \lambda A_1 + A_0)\mathbf{F} = 0. \tag{41}$$

For this polynomial eigenvalue problem, the target smallest positive eigenvalues are embedded in the interior of the spectrum of the matrix polynomial (41). This is merely an algebraic artifact as we have approximated the effective mass using rational functions of the unknown eigenvalues, hence raised the degree of the polynomial and produced extra roots located on the negative real axis. These negative eigenvalues are non-physical and should not be taken as solutions.

To compute the smallest positive eigenvalue for the linear eigenvalue problem (40), variants of the rational Krylov subspace method [24,34,35] can be used to compute the eigenpairs. Theoretically, these methods can also be applied to the quintic polynomial (41) written as an enlarged linear eigenvalue problem

$$\begin{bmatrix} 0 & I & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & I \\ A_0 & A_1 & A_2 & A_3 & A_4 \end{bmatrix} \begin{bmatrix} \mathbf{F} \\ \lambda\mathbf{F} \\ \lambda^2\mathbf{F} \\ \lambda^3\mathbf{F} \\ \lambda^4\mathbf{F} \end{bmatrix} = \lambda \begin{bmatrix} I & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 \\ 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & -A_5 \end{bmatrix} \begin{bmatrix} \mathbf{F} \\ \lambda\mathbf{F} \\ \lambda^2\mathbf{F} \\ \lambda^3\mathbf{F} \\ \lambda^4\mathbf{F} \end{bmatrix}. \tag{42}$$

However, the relevant eigenvalues are the smallest positive ones and they are located in the interior of the matrix eigenvalue spectrum. To solve them successively by these methods, the classical shift-and-invert technique is much too expensive for large scale problems. Moreover, the enlarged matrix is five times larger than the A_i's. and does not have a good structure as the A_i's do. The condition number for the enlarged matrix can be significantly larger than those of the A_i's since it has a large set of admissible perturbations [42]. As a consequence, all these approach are not favorable in terms of accuracy and efficiency. To bypass these difficulties, it is desirable to have numerical methods that deal with the eigenvalue problems (40) and (41) directly by such as the Jacobi–Davidson methods. It is thus straightforwardly to use the linear Jacobi–Davidson method [38] for the linear eigenvalue problem (40). However, rare numerical results were reported for the quintic eigenvalue problem (41), while variants of the Jacobi–Davidson methods are considered for the generalized linear [37], quadratic [19,39], and cubic [43] eigenvalue problems.

Besides, to compute successively all other desired eigenvalues, deflation or locking schemes are necessary. The implicit deflation techniques based on the Schur forms is known to have good performance for linear eigenvalue problems [1, Section 4.7 and 8.4]. However, it is not clear how to incorporate the implicit deflation technique in a quintic polynomial eigenvalue problem since the Schur form is not defined for a quintic polynomial matrix in general. Alternatively, an explicit deflation scheme is proposed in [43] for cubic eigenvalue problems. Once the smallest eigenvalue is obtained, the scheme transforms the computed eigenvalue to infinity and keeps all other eigenvalues unchanged. The second eigenvalue thus becomes the smallest eigenvalue of the newly transformed eigenvalue problem, which can be solved by the cubic Jacobi–Davidson method. The technique is then applied repeatedly until all desired eigenvalues are determined. In addition to the deflation schemes, Meerbergen [30] proposed a quadratic eigensolver using the locking and restarting scheme based on the Schur form of the linearized problem, which is similar to (41). The method gives a linkage between the methods for solving the quadratic and the linearized eigenvalue problems.

Here we propose a general polynomial Jacobi–Davidson algorithm, shown in Fig. 6, to compute all the desired eigenvalues for the problem (9). This algorithm is based on a polynomial Jacobi–Davidson method modified directly from the quadratic Jacobi–Davidson method illustrated in [2, Section 9.2] and a locking technique similar to the one mentioned in [30]. This algorithm is capable of calculating the smallest positive eigenvalue first and then computing successively all other desired eigenvalues by suitably choosing the orthonormal searching space $V = [V_\mathbf{F}, V_{\text{ini}}]$:

- The algorithm computes the smallest positive eigenvalue λ_1 and the associated eigenvector $\mathbf{F}_1$ by the general Jacobi–Davidson method first. In this stage, $V_\mathbf{F}$ is empty, V_{ini} is any matrix satisfying $V_{\text{ini}}^{\mathrm{T}} V_{\text{ini}} = I$ (in Step (1) of Fig. 6).

- To compute the second smallest positive eigenvalue λ_2 ($j = 2$ in Step (2)), the algorithm normalizes the approximate eigenvalue $\mathbf{F}_1$ and adds it to the initial search space $V_{\mathbf{F}}$. That is, the algorithm computes $V_{\mathbf{F}} = \frac{\mathbf{F}_1}{\|\mathbf{F}_1\|}$ and chooses an orthonormal matrix V_{ini} such that $[V_{\mathbf{F}}, V_{\mathrm{ini}}]^{\mathrm{T}}[V_{\mathbf{F}}, V_{\mathrm{ini}}] = I$. By doing so, the first eigenvalue λ_1 will be included (or "locked") in the eigenpairs computed by

$$(V^{\mathrm{T}}\mathbf{A}(\lambda)V)s = 0 \tag{43}$$

in Step (2.2.i). Therefore, the algorithm chooses the second smallest positive eigenvalue from the ones in (43).

- The algorithm computes $\lambda_{\hat{j}}$ in a similar way. The algorithm sets $V_{\mathbf{F}}$ to be an orthonormal basis of the eigenspace spanned by $\mathbf{F}_1, \ldots, \mathbf{F}_{\hat{j}-1}$ and then choose an orthonormal matrix V_{ini} satisfying $[V_{\mathbf{F}}, V_{\mathrm{ini}}]^{\mathrm{T}}[V_{\mathbf{F}}, V_{\mathrm{ini}}] = I$. The eigenvalues λ_1 through $\lambda_{\hat{j}-1}$ will be locked in the eigenpairs computed in Eq. (43). The algorithm thus chooses the $\hat{j}$th smallest positive eigenvalue from the ones obtained in (43) as an approximation of $\lambda_{\hat{j}}$.

To conclude this algorithm, we note that the correction equation

$$\left(I - \frac{\boldsymbol{p}\boldsymbol{u}^*}{\boldsymbol{u}^*\boldsymbol{p}}\right)\mathbf{A}(\theta)(I - \boldsymbol{u}\boldsymbol{u}^*)\boldsymbol{t} = -\boldsymbol{r} \tag{44}$$

in Step (2.2.v) is solved approximately by computing

$$\boldsymbol{t} = -M_A^{-1}\boldsymbol{r} + \varepsilon M_A^{-1}\boldsymbol{p} \quad \text{with } \varepsilon = \frac{\boldsymbol{u}^*M_A^{-1}\boldsymbol{r}}{\boldsymbol{u}^*M_A^{-1}\boldsymbol{p}}. \tag{45}$$

The matrix M_A is a preconditioner of $\mathbf{A}(\theta)$. For example, in the SSOR preconditioning scheme [40] used in our numerical experiments,

$$\mathbf{A}(\theta) \approx M_{\mathbf{A}} = (D - \omega L)D^{-1}(D - \omega U),$$

where ω is a scalar, $\mathbf{A}(\theta) = D - L - U$ with $D = \mathrm{diag}(\mathbf{A}(\theta))$, L and U are strictly lower and upper triangular of $\mathbf{A}(\theta)$.

Remark. The polynomial Jacobi–Davidson algorithm outlined in Fig. 6 can be applied to general matrix polynomial eigenvalue problems. We have implemented and tested it up to $\tau = 5$ at very large scale (32 million unknowns) and are quite satisfied with its performance. As analyzed in Section 4.1, the performance of our scheme is related to the structures of the matrices A_i. The algorithm did not make use of any special properties related to the degree of the polynomial and we expect it to work as well for higher degree polynomial eigenvalue problems that arises from discretizing (7) over regular or truncated pyramid dots.

4.1. Analysis of the quintic polynomial eigenvalue problem

We have found the performance of the algorithm quite satisfactory for both the linear and quintic polynomial eigenvalue problems. See Section 5 for detail. Here we provide a heuristic explanations of the rapid convergence of the algorithms based on the structures of the coefficient matrices A_i. Fig. 5 shows the sparsity patterns of the matrices A_0 through A_5 with $(L, M, N) = (8, 8, 6)$.

For the constant effective mass model (40), it is easy to verify that A_0 is symmetric and diagonal dominant.

For the non-parabolic effective mass approximation (41), the matrices A_i have the following properties:

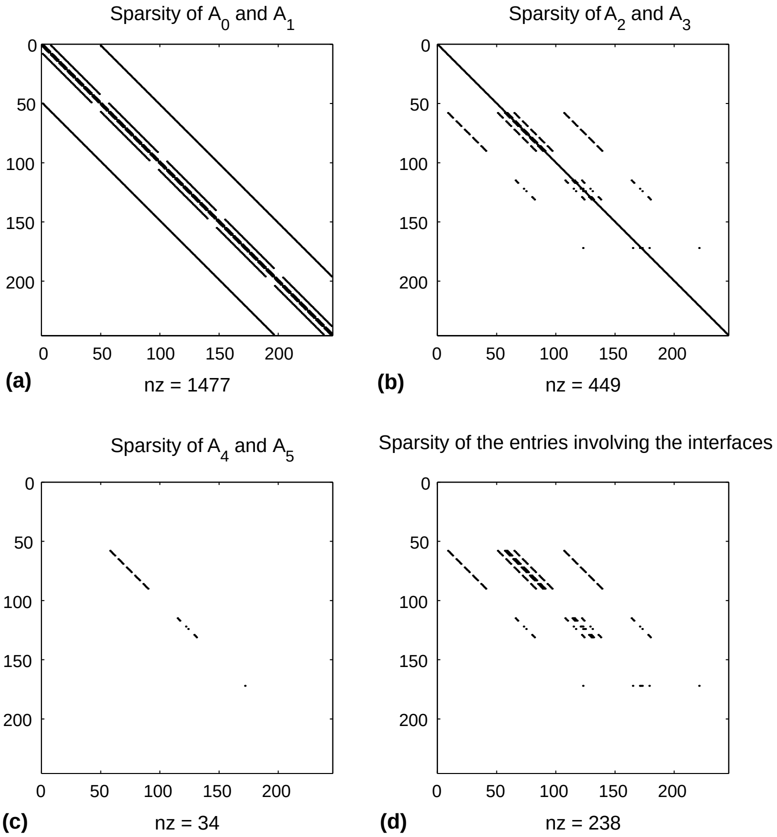

Fig. 5. Sparsity patterns of the matrices A_0 through A_5 are presented in part (a)–(c). In part (d), only the entries involving the interfaces are plotted. The notation "nz" in the figure stands for the number of non-zero entries in the matrices.

4.1.1. Symmetry

The matrices A_i, $i = 0, 1, 2, 3$, are symmetric except the entries involving the grids on the interface. A_4 and A_5 are diagonal. See Fig. 5(c).

4.1.2. Diagonal dominance

The matrix A_0 is diagonal dominant. The matrices A_1, A_2, and A_3 are *almost* diagonal dominant except the rows involving the grids on the interface. These rows of A_2 and A_3 are indicated in part (d) of Fig. 5. The matrices A_4 and A_5 are diagonal dominant since they are diagonal matrices.

4.1.3. Asymptotic structure

- The structures of A_0 and A_1 mentioned above are independent of h.
- The entries of A_2 and A_3 are of $O(h^2)$ except those associated with the interface grids. For the matrices A_2 and A_3, we observe the main diagonal of the matrices are formed by $O(h^2)$ entries, except the ones associated with the interface conditions. The rows associated with these interface grids has non-zero $O(1)$ entries on the main diagonal and six off-diagonals. The sparsity of A_2 and A_3 are shown in Fig. 5(b).
- The entries of the diagonal matrices A_4 and A_5 are all $O(h^2)$.

```
(0) Given $\mathbf{A}(\lambda) = \sum_{i=0}^{\tau} \lambda^i A_i$ and the number of desired eigenvalues $\sigma$.
(1) Initialize $V = [V_{ini}]$ as an orthonormal matrix.
(2) For $j = 1, 2, ..., \sigma$
    (2.1) Compute $W_k = A_k V$ and $M_k = V^* W_k$, for $k = 0, 1, ..., \tau$.
    (2.2) Iterate until convergence (the Jacobi–Davidson iteration)
        (i)    Compute all the eigenpairs of the small size problem
               $(\sum_{i=0}^{\tau} \theta^i M_i)\boldsymbol{s} = 0$.
        (ii)   Select $\theta$ as the $j$th smallest positive eigenvalue and
               let $\boldsymbol{s}$ be the associated normalized eigenvector.
        (iii)  Compute $\boldsymbol{u} = V\boldsymbol{s}$, $\boldsymbol{p} = \mathbf{A}'(\theta)\boldsymbol{u}$, and $\boldsymbol{r} = \mathbf{A}(\theta)\boldsymbol{u}$.
        (iv)   If $(||\boldsymbol{r}||_2 < \varepsilon)$, Set $\lambda_j = \theta$, $\mathbf{F}_j = \boldsymbol{u}$, Goto Step (2.3).
        (v)    Solve (approximately) a $\boldsymbol{t} \perp \boldsymbol{u}$ from
               $(I - \frac{\boldsymbol{p}\boldsymbol{u}^*}{\boldsymbol{u}^*\boldsymbol{p}})\mathbf{A}(\theta)(I - \boldsymbol{u}\boldsymbol{u}^*)\boldsymbol{t} = -\boldsymbol{r}$.
        (vi)   Orthogonalize $\boldsymbol{t}$ against $V$; Set $\boldsymbol{v} = \boldsymbol{t}/||\boldsymbol{t}||_2$.
        (vii)  Compute $\boldsymbol{w}_k = A_k\boldsymbol{v}$, $M_k = \begin{bmatrix} M_k & V^*\boldsymbol{w}_k \\ \boldsymbol{v}^*W_k & \boldsymbol{v}^*\boldsymbol{w}_k \end{bmatrix}$,
               and $W_k = [W_k, \boldsymbol{w}_k]$, for $k = 0, 1, ..., \tau$.
        (viii) Expand $V = [V, \boldsymbol{v}]$
    (2.3) Orthogonalize $\mathbf{F}_j$ against $V_{\mathbf{F}}$; Compute $\mathbf{F}_j = \frac{\mathbf{F}_j}{||\mathbf{F}_j||}$;
          Update $V_{\mathbf{F}} = [V_{\mathbf{F}}, \mathbf{F}_j]$.
    (2.4) Choose an orthonormal matrix $V_{ini} \perp V_{\mathbf{F}}$; Set $V = [V_{\mathbf{F}}, V_{ini}]$
End for
(3) Output the approximated eigenpairs $(\lambda_j, \mathbf{F}_j)$ for $j = 1, ..., \sigma$.
```

Fig. 6. The polynomial Jacobi–Davidson algorithm designed to compute the σ smallest positive eigenvalues and the associated eigenvectors of polynomial eigenvalue problems (9).

In summary, as h goes to zero, the quintic matrix polynomial can be viewed as a small perturbation of the generalized linear eigenvalue problem

$$A_0\mathbf{F} = -\lambda A_1\mathbf{F} + \mathrm{O}(h^2) + B,$$

with A_0 being diagonal dominant and nearly symmetric. Here B is a low rank matrix. The only non-zero entries are those associated with interface grids.

It is known [23] that for symmetric generalized linear eigenvalue problems, the more diagonally dominant the matrix is, the less iteration numbers Davidson's method takes. For the symmetric linear eigenvalue problem $Ax = \lambda x$, Morgan and Scott [31] also provided an analysis for local and global convergence of Davidson's method based on diagonal preconditioners and the gap ratio of the desired eigenvalue. More recently, Sleijpen et al. [37] studied convergence of Jacobi–Davidson type methods for generalized eigenvalue problems. Their main concern is the projection that is used to reduce the given eigenvalue problem to a smaller one. They showed that a proper choice for the projection operator can lead to quadratic convergence.

From our working experience and the related works mentioned above, we believe that the symmetry and diagonal dominance of the matrices in the matrix polynomial (41) plays an essential roles in the rapid convergence.

5. Numerical results

In this section, we report several numerical experiments to access the performance of our scheme. We have used true physical parameters in our simulations. The width of the QD base is 12.4 nm and the height of the QD is 6.2 nm. This assumption was made to be approximately comparable with the experimental results reported in [9]. The size of the cuboid matrix is taken to be 24.8 nm × 24.8 nm × 18.6 nm. The cuboid matrix is partitioned into L, M and N meshes in each direction with $\Delta x = 24.8/L$, $\Delta y = 24.8/M$ and $\Delta z = 18.6/N$. Since we have imposed homogeneous Dirichlet boundary condition, the total number of unknowns, or the dimension of A_i's, is therefore $(L-1)\times(M-1)\times(N-1)$.

For the constant effective mass model, the effective mass for the QD and the cuboid matrix are $0.024m_e$ and $0.067m_e$, respectively. For the non-parabolic effective mass approximation, we used the following parameters [27]: $P_1 = 0.8503$, $g_1 = 0.42$, $\delta_1 = 0.48$, $V_1 = 0.0$, $P_2 = 0.8878$, $g_2 = 1.52$, $\delta_2 = 0.34$, and $V_2 = 0.7$.

The iteration process within the Jacobi–Davidson subroutine terminates when the residual of the eigenvalue problem (9) reaches 1.0×10^{-10} for the computed eigenpairs.

5.1. Rate of convergence for the finite volume discretization

We first check the convergence behavior of the discretization scheme. As shown in Section 2, the local truncation error is $O(h)$ on the interface grids and $O(h^2)$ elsewhere. From standard error estimate, we expect the global error to be $O(h^2)$ and this is indeed verified in our tests.

We choose $\Delta x = \Delta y = \Delta z = h$ for convenience and compute the first three eigenvalues (i.e. λ_1, λ_2, and λ_3) for various h. As mesh refines, the rate of convergence on eigenvalues is measured by

$$\log_2\left(\frac{\lambda^{(4h)} - \lambda^{(2h)}}{\lambda^{(2h)} - \lambda^{(h)}}\right),$$

where $\lambda^{(h)}$ is the eigenvalue computed with mesh size h. The result is summarized in Table 1.

Similarly, we measure the rate of convergence on eigenvectors by computing

$$\log_2\left(\frac{\|\mathbf{F}_{4h}^{(4h)} - \mathbf{F}_{4h}^{(2h)}\|_2}{\|\mathbf{F}_{4h}^{(2h)} - \mathbf{F}_{4h}^{(h)}\|_2}\right),$$

where $\mathbf{F}_{4h}^{(h)}$ is the eigenvector computed with mesh size h and then restricted to the grids with mesh size $4h$. Thus $\mathbf{F}_{4h}^{(h)}$, $\mathbf{F}_{4h}^{(2h)}$ and $\mathbf{F}_{4h}^{(2h)}$ are all defined on the same grid points. The notation $\|\cdot\|_2$ stands for the L_2 norm of

Table 1
Computed eigenvalue results of the finite volume scheme over various mesh size

(L,M,N)	Matrix dimension	λ_1	Rate	λ_2	Rate	λ_3	Rate
Constant effective mass model							
(16,16,12)	2475	0.4226	–	0.6527	–	0.6527	–
(32,32,24)	22,103	0.4001	–	0.6423	–	0.6423	–
(64,64,48)	186,543	0.3934	1.744	0.6391	1.708	0.6391	1.708
(128,128,96)	1,532,255	0.3916	1.905	0.6383	1.866	0.6383	1.866
(256,256,192)	12,419,775	0.3911	1.954	0.6380	1.912	0.6380	1.912
Non-parabolic effective mass model							
(16,16,12)	2475	0.4314	–	0.6101	–	0.6101	–
(32,32,24)	22,103	0.4204	–	0.6022	–	0.6022	–
(64,64,48)	186,543	0.4173	1.813	0.5999	1.793	0.5999	1.793
(128,128,96)	1,532,255	0.4165	1.948	0.5993	1.938	0.5993	1.938
(256,256,192)	12,419,775	0.4163	1.986	0.5991	1.983	0.5991	1.983

Table 2
Rate of convergence on the computed eigenvectors over various mesh size

(L,M,N)	Constant effective mass model		Non-parabolic effective mass model	
	$\mathbf{F}_1$ Rate	$\mathbf{F}_2$ Rate	$\mathbf{F}_1$ Rate	$\mathbf{F}_2$ Rate
(16,16,12)	–	–	–	–
(32,32,24)	–	–	–	–
(64,64,48)	1.622	1.679	2.042	2.152
(128,128,96)	1.873	1.884	1.993	2.011
(256,256,192)	1.953	1.937	1.997	2.001

a vector. The result for the eigenvectors associated with the first two eigenvalues is summarized in Table 2. The result for the third eigenvector is not shown, since the second and the third eigenvalues are the same and the associated eigenvectors are thus identical up to a unitary transformation.

To verify the computed eigenvalues in Table 1 are correct, we use the `polyeig` function in Matlab [41] to compute the full matrix eigenvalue spectrum of a small problem with non-parabolic effective mass approximation and $(L,M,N) = (12,12,9)$. The function `polyeig` solves the enlarged system (42) and only works for small size problems. Fig. 7 shows the spectrum, among which are the three desired eigenvalues, 0.4408, 0.6170, and 0.6170. The results are identical with the ones computed by our algorithm up to the 12 digits.

The computed wave function associated with the first two smallest positive eigenvalues are shown in Fig. 8 with $(L,M,N) = (32,32,24)$.

5.2. Performance of the algorithm for large scale eigenvalue problems

Here we measure the performance of the algorithm by the number of Jacobi–Davidson iterations taken in Step (2.2) of the algorithm listed in Fig. 6 for each of the eigenvalue and total CPU time.

We first conduct the numerical simulation on a ordinary desktop equipped with a Pentium IV 1.8 GHz CPU and 1 gigabyte main memory. The result for $(L,M,N) = (128,128,96)$ or 1,532,255 unknowns is listed in Table 4. This simulation consumes about 800 megabytes memory.

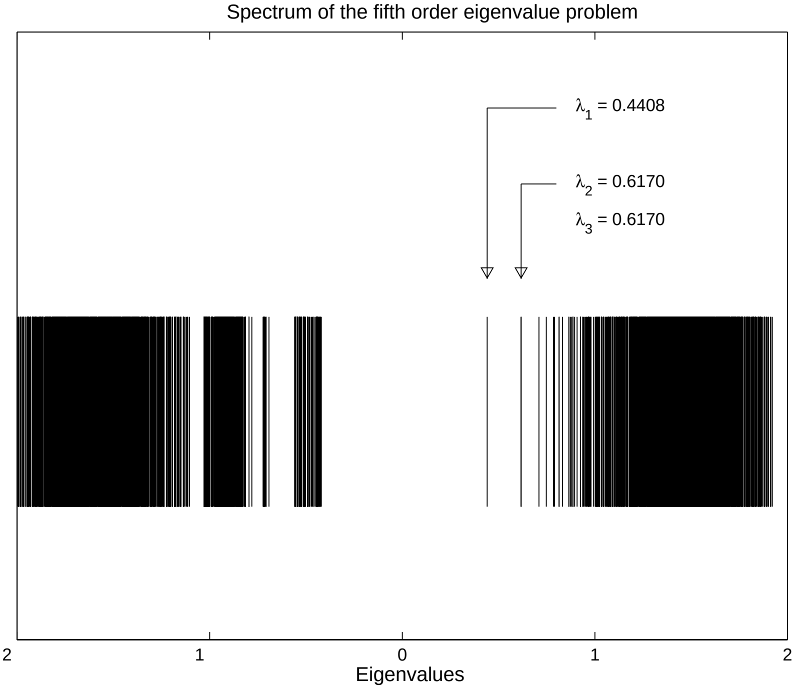

Fig. 7. The matrix eigenvalue spectrum of a quintic polynomial eigenvalue problem with $A_i \in \mathbb{R}^{968\times 968}$, $i = 0,\ldots,5$ computed by the function `polyeig` in Matlab. The desired eigenvalues, λ_1, λ_2, and λ_3, are located in the interior of the spectrum.

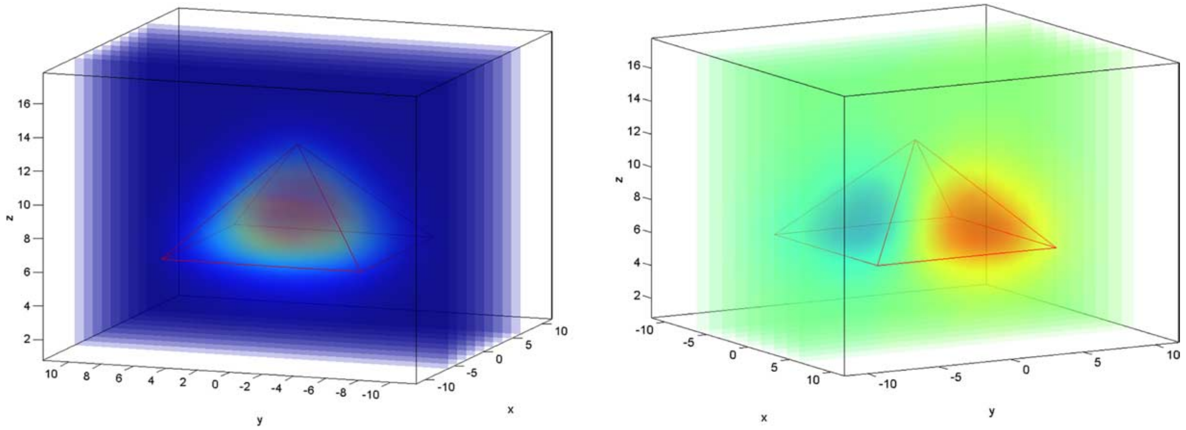

Fig. 8. Wave functions (eigenvectors) associated with the ground state and the first excited state energies (the first two smallest positive eigenvalue) for the 3D pyramid QD with $(L,M,N) = (32,32,24)$.

From Tables 1 and 4, we can asses the applicability of our scheme on practical problems. For example, a cuboid matrix with dimensions 49.6 nm × 49.7 nm × 37.2 nm discretized using $(L,M,N) = (128,128,96)$ points can be simulated easily on an ordinary PC as mentioned above. A typical pyramid QD is about 12.4 nm × 12.4 nm × 6.2 nm in dimension and the cuboid matrix can easily accommodate several QDs

Table 3
Computational results of a eigenvalue problem with size 32,401,863

	Constant effective mass model (linear eigenvalue problem)			Non-parabolic effective mass model (quintic eigenvalue problem)		
	Value	Item number	CPU time (s)	Value	Item number	CPU time (s)
λ_1	0.3910	138	5852	0.4162	84	4856
λ_2	0.6380	133	5354	0.5991	74	4835
λ_3	0.6380	220	8511	0.5991	113	8280

For the first three eigenvalues, the table shows the computed eigenvalues, numbers of iterations, and total CPU time in seconds.

Table 4
Computational results of a eigenvalue problem with 1,532,255 unknowns on an desktop with 1.8 GHz Pentium 4 CPU and 1 gigabyte of main memory

	Constant effective mass model (linear eigenvalue problem)			Non-parabolic effective mass model (quintic eigenvalue problem)		
	Value	Item number	CPU time (s)	Value	Item number	CPU time (s)
λ_1	0.3916	72	278.0	0.4165	47	447.4
λ_2	0.6383	72	284.3	0.5993	45	460.7
λ_3	0.6383	125	521.4	0.5993	50	544.7

inside. The grid size corresponds to $h = 0.3875$, same as the (64,64,48) simulation in Table 1 (a 24.8 nm × 24.8 nm × 12.6 nm cuboid), which resulted relative errors ranging from 0.2% to 0.5%.

To find out the relation between the number of Jacobi–Davidson iteration and the size of the matrices A_i, we repeat the simulation on a decent workstation with a 1.0 GHz Intel Itanium II CPU and 12 gigabytes of main memory. The result is given in Table 5 and plotted in Fig. 9. The number of iteration increases very slowly with respect to the dimension of the matrix (number of unknowns).

We have pushed the problem size up to $(L, M, N) = (352, 352, 264)$ or 32,401,863 unknowns, which consumed about 8 gigabytes of main memory. We have not exploited the mirror symmetries of the pyramid QD, which will further reduce the number of unknowns by a factor of 1/8. In Table 3, we list the computed eigenvalues, number of Jacobi–Davidson iteration and total CPU time in seconds. It is quite remarkable that for a polynomial eigenvalue problem with more than 32 millions unknowns, the algorithm took only a couple of hundreds iterations and several thousands seconds of CPU time.

Table 5
Number of Jacobi–Davidson iteration for the first 3 eigenvalues over various mesh size h

h (nm)	Matrix dimension	Constant model				Non-parabolic model			
		λ_1	λ_2	λ_3	Average	λ_1	λ_2	λ_3	Average
1.55×10^0	2475	34	38	45	39	36	41	29	35
7.75×10^{-1}	22,103	46	48	62	52	44	44	27	38
3.88×10^{-1}	186,543	54	57	84	65	44	43	42	43
1.94×10^{-1}	1,532,255	72	72	117	87	47	45	57	50
9.69×10^{-2}	12,419,775	108	106	195	136	66	58	86	70
7.05×10^{-2}	32,401,863	138	133	220	163	84	74	113	90

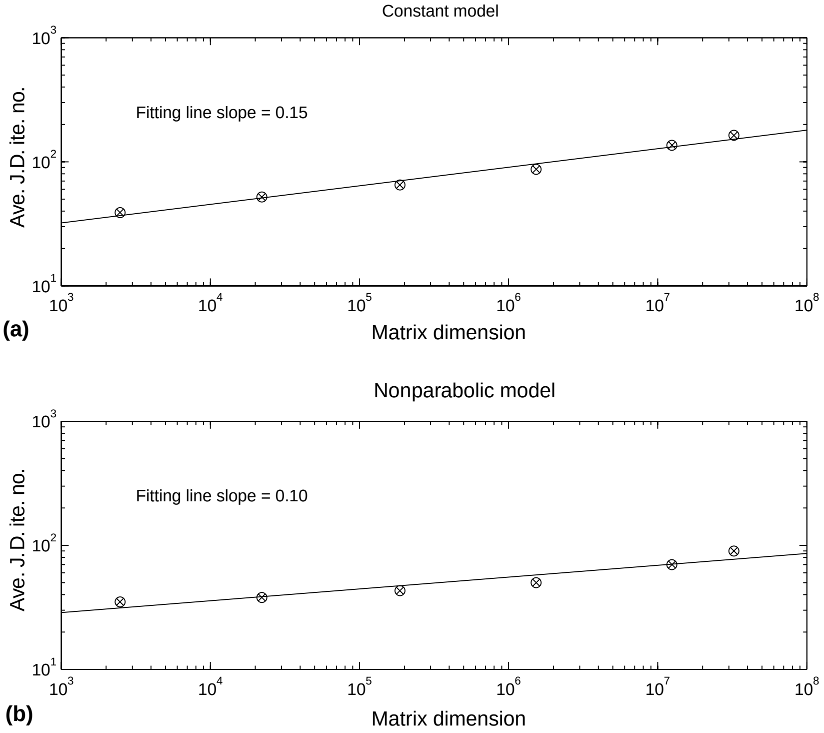

Fig. 9. The average Jacobi–Davidson iteration numbers versus the size of the matrices A_i.

6. Conclusion

In this article, we propose a 2nd order finite volume scheme to discretize the Schrödinger equation with uniform mesh. The discretization leads to large scale linear and non-linear eigenvalue problems. The relevant eigenvalues, which are embedded in the interior of the matrix eigenvalue spectrum, are then solved successively using a Jacobi–Davidson based method. Numerical evidence shows that the overall scheme is stable and extremely efficient. The number of iterations grows very slowly with the matrix size. We have tested the quintic polynomial eigenvalue problems with up to 32 millions variables and the eigenvalues converge within a little over one hundred iterations. We have also analyzed the structures of the matrices and attribute the superior performance to the symmetry and diagonal dominance of the matrices.

Acknowledgements

The authors thank Weihua Wang for many helpful discussions. The authors are also grateful to the anonymous referees for the helpful comments and suggestions. This work is partially supported by the National Science Council and the National Center for Theoretical Sciences in Taiwan.

Appendix A. Finite difference interpretation of the discretization

Here we re-derive the finite volume scheme (28) as a finite difference scheme that incorporates the jump condition automatically without enforcing them explicitly.

Consider a grid point on the left hypotenuse of the 2D triangle domain (Fig. 2(d)). We have

$$u_{i+1,j} - u_{i,j} = (\Delta x)u_x^- + \frac{\Delta x^2}{2}u_{xx}^- + \mathrm{O}(h^3) \tag{A.1}$$

and

$$u_{i,j-1} - u_{i,j} = -(\Delta y)u_y^- + \frac{\Delta y^2}{2}u_{yy}^- + \mathrm{O}(h^3), \tag{A.2}$$

respectively. Here the sign in the superscript refers to the one sided limit from the interior (−) and exterior of the interface rather than the direction of the coordinate axis.

From (A.1) and (A.2), we have

$$\frac{\alpha^-}{\Delta x\Delta y}\left[\frac{\Delta y}{\Delta x}(u_{i+1,j} - u_{i,j}) + \frac{\Delta x}{\Delta y}(u_{i,j-1} - u_{i,j})\right] = \frac{\alpha^-}{\Delta x\Delta y}\left[(\Delta y)u_x^- . - (\Delta x)u_y^- + \frac{\Delta x\Delta y}{2}(u_{xx}^- + u_{yy}^-)\right] + \mathrm{O}(h). \tag{A.3}$$

The first two terms on the right hand side can be further reduced to

$$(\Delta y)u_x^- - (\Delta x)u_y^- = \sqrt{\Delta x^2 + \Delta y^2}\left(\frac{(\Delta y, -\Delta x)\cdot\nabla^- u}{\sqrt{\Delta x^2 + \Delta y^2}}\right) = \sqrt{(\Delta x)^2 + (\Delta y)^2}u_n^-, \tag{A.4}$$

where u_n^- denotes the interior limit of the normal derivative. Similarly,

$$u_{i-1,j} - u_{i,j} = -(\Delta x)u_x^+ + \frac{\Delta x^2}{2}u_{xx}^+ + \mathrm{O}(\Delta x^3), \tag{A.5}$$

$$u_{i,j+1} - u_{i,j} = (\Delta y)u_y^+ + \frac{\Delta y^2}{2}u_{yy}^+ + \mathrm{O}(\Delta y^2) \tag{A.6}$$

and we have

$$\begin{aligned}
&\frac{\alpha^+}{\Delta x\Delta y}\left[\frac{\Delta y}{\Delta x}(u_{i-1,j} - u_{i,j}) + \frac{\Delta x}{\Delta y}(u_{i,j+1} - u_{i,j})\right]\\
&\quad = \frac{\alpha^+}{\Delta x\Delta y}\left[-(\Delta y)u_x^+ + (\Delta x)u_y^+ + \frac{\Delta x\Delta y}{2}(u_{xx}^+ + u_{yy}^+)\right] + \mathrm{O}(\Delta x)\\
&\quad = \frac{\alpha^+}{\Delta x\Delta y}\left(-\sqrt{(\Delta x)^2 + (\Delta y)^2}u_n^+ + \frac{\Delta x\Delta y}{2}(u_{xx}^+ + u_{yy}^+)\right) + \mathrm{O}(\Delta x).
\end{aligned} \tag{A.7}$$

From Eqs. (A.3) and (A.7), we conclude that

$$\begin{aligned}
&-\frac{1}{\Delta x^2}\left[\alpha^- u_{i+1,j} - (\alpha^- + \alpha^+)u_{i,j} + \alpha^+ u_{i-1,j}\right] - \frac{1}{\Delta y^2}\left[\alpha^- u_{i,j-1} - (\alpha^- + \alpha^+)u_{i,j} + \alpha^+ u_{i,j+1}\right]\\
&\quad = -\frac{1}{2}\alpha^-(u_{xx}^- + u_{yy}^-) - \frac{1}{2}\alpha^+(u_{xx}^+ + u_{yy}^+) + \mathrm{O}(h),
\end{aligned} \tag{A.8}$$

where we have used the interface condition (13). After substituting the Schrödinger equation (1) into the right hand side of (A.8), we have re-derived the finite volume discretization (19)

$$-\frac{1}{\Delta x^2}\left[\alpha^- u_{i+1,j} - (\alpha^- + \alpha^+)u_{i,j} + \alpha^+ u_{i-1,j}\right] - \frac{1}{\Delta y^2}\left[\alpha^- u_{i,j-1} - (\alpha^- + \alpha^+)u_{i,j} + \alpha^+ u_{i,j+1}\right]R$$
$$= -\frac{1}{2}(V^- + V^+)u_{i,j} + \lambda u_{i,j} + \mathrm{O}(h). \tag{A.9}$$

The finite difference interpretation on other interface points can be derived similarly, we omit the details.

Remark. The local Taylor expansions (A.1), (A.2), (A.5) and (A.6) results in O(h) local truncation errors at the interface grids. If we include higher order derivatives in the local Taylor expansions above, combined with higher order discretization at interior and exterior grids, it is straightforward to obtain higher order finite difference schemes with similar matrix structures. This topic is currently under investigation and the results will be reported elsewhere.

References

[1] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, H. van der Vorst, Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide, SIAM, Philadelphia, 2000.

[2] Z. Bai, G. Sleijpen, H. van der Vorst, Nonlinear eigenvalue problems, in: Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, H. van der Vorst (Eds.), Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide, SIAM, Philadelphia, 2000 (Chapter 9).

[3] G. Bastard, Wave Mechanics Applied to Semiconductor Heterostructures, John Wiley and Sons, New York, 1991.

[4] G.L. Bir, G.E. Pikus, Symmetry and Strain-Induced Effects in Semiconductors, Wiley, New York, 1974.

[5] J. Bramble, J. King, A finite element method for interface problems in domains with smooth boundaries and interfaces, Adv. Comput. Math. 6 (1996) 109–138.

[6] G. Burkard, D. Loss, D.P. DiVincenzo, Couple quantum dots as quantum gates, Phys. Rev. B 59 (1999) 2070–2078.

[7] Z. Chen, J. Zou, Finite element methods and their convergence for elliptic and parabolic interface problems, Numer. Math. 79 (1998) 175–202.

[8] S. Chuang, Physics of Optoelectronic Devices, Wiley, New York, 1995.

[9] M.A. Cusack, P.R. Briddon, M. Jaros, Electronic structure of InAs/GaAs self-assembled quantum dots, Phys. Rev. B 54 (1996) 2300–2303.

[10] D. El-Moghraby, R.G. Johnson, P. Harrison, Calculating modes of quantum wire and dot systems using a finite differencing technique, Comp. Phys. Commun. 150 (2003) 235–246.

[11] R. Ewing, Z. Li, T. Lin, Y. Lin, The immersed finite volume element methods for the elliptic interface problems, Math. Comput. Simul. 50 (1999) 63–76.

[12] L.R.C. Fonseca, J.L. Jimenez, J.P. Leburton, Electronic coupling in InAs/GaAs self-assembled stacked double-quantum-dot systems, Phys. Rev. B 58 (1998) 9955–9960.

[13] L.R.C. Fonseca, J.L. Jimenez, J.P. Leburton, Erratum: electronic coupling in InAs/GaAs self-assembled stacked double-quantum-dot systems, Phys. Rev. B 58 (1998) 9955–9960 [Phys. Rev. B 58 (1998) 9955].

[14] L.R.C. Fonseca, J.L. Jimenez, J.P. Leburton, R. Martin, Self-consistent calculation of the electronic structure and electron-electron interaction in self-assembled InAs–GaAs quantum dot structures, Phys. Rev. B 57 (1998) 4017–4026.

[15] J.W. Gray, D. Childs, S. Malik, P. Siverns, C. Roberts, P.N. Stavrinou, M. Whitehead, R. Murray, G. Parry, Quantum dot resonant cavitylight emitting diode operating near 1300 nm, Electron. Lett. 35 (1999) 242.

[16] L. Harris, D.J. Mowbray, M.S. Skolnick, M. Hopkinson, G. Hill, Emission spectra and mode structure of InAs/GaAs self-organized quantum dot lasers, Appl. Phys. Lett. 73 (1998) 969–971.

[17] P. Harrison, Quantum Wells, Wires, and Dots: Theoretical and Computational Physics, John Wiley & Sons, New York, 2000.

[18] R. Heitz, M. Veit, N.N. Ledentsov, A. Hoffmann, D. Bimberg, V.M. Ustinov, P.S. Kopév, Z.I. Alferov, Energy relaxation by multiphonon processes in InAs/GaAs quantum dots, Phys. Rev. B 56 (1997) 10435–10445.

[19] M.E. Hochstenbach, H.A. van der Vorst, Alternatives to the Rayleigh quotient for the quadratic eigenvalue problem, Technical Report 1212, Department of Mathematics, University of Utrecht, P.O. Box 80.010, NL-3508 TA Utrecht, The Netherlands, 2001.

[20] Y.-L. Hwang, W.-C. Wang. A monotone jump condition capturing scheme for elliptic interface problems on arbitrary domains, Preprint.

[21] G. Iannaccone, A. Trellakis, U. Ravaioli, Simulation of a quantum-dot flash memory, J. Appl. Phys. 84 (9) (1998) 5032–5036.

[22] L. Jacak, P. Hawrylak, A. Wójs, Quantum Dots, Springer, Berlin, 1998.

[23] T.Z. Kalamboukis, Davidson's algorithm with and without perturbation corrections, J. Phys. A: Math. General 13 (1) (1980) 57–62.

[24] R.B. Lehoucq, K. Meerbergen, Using generalized Cayley transformations within an inexact rational krylov sequence method, SIAM J. Matrix Anal. Appl. 20 (1998) 131–148.

[25] R.J. LeVeque, Z. Li, Immersed interface methods for stokes flow with elastic boundaries or surface tension, SIAM J. Sci. Comput. 18 (1997) 709–735.

[26] Y. Li, J. Liu, O. Voskoboynikov, C.P. Lee, S.M. Sze, Electron energy level calculations for cylindrical narrow gap semiconductor quantum dot, Comput. Phys. Commun. 140 (2001) 399–404.

[27] Y. Li, O. Voskoboynikov, C.P. Lee, S.M. Sze, Computer simulation of electron energy levels for different shape InAs/GaAs semiconductor quantum dots, Comput. Phys. Commun. (141) (2001) 66–72.

[28] S. Maimon, E. Finkman, G. Bahir, S.E. Schacham, J.M. Garcia, P.M. Petroff, Intersublevel transitions in InAs/GaAs quantum dots infrared photodetectors, Appl. Phys. Lett. 73 (1998) 2003–2005.

[29] G. Medeiros-Ribeiro, J.M. Garcia, P.M. Petroff, Charging dynamics of InAs self-assembled quantum dots, Phys. Rev. B 56 (1997) 3609–3612.

[30] K. Meerbergen, Locking and restarting quadratic eigenvalue solvers, SIAM J. Sci. Comput. 22 (5) (2001) 1814–1839.

[31] R.B. Morgan, D.S. Scott, Generalizations of Davidson's method for computing eigenvalues of sparse symmetric matrices, SIAM J. Sci. Stat. Comput. 7 (3) (1986) 817–825.

[32] F.M. Peeters, V.A. Schweigert, Two-electron quantum disks, Phys. Rev. B 53 (1996) 1468–1474.

[33] C. Pryor, Eight-band calculations of strained InAs/GaAs quantum dots compared with one-, four-, and six-band approximations, Phys. Rev. B 57 (1998) 7190–7195.

[34] A. Ruhe, The rational krylov: a practical algorithm for large sparse nonsymmetric matrix pencils, SIAM J. Sci. Comput. 19 (1998) 1535–1551.

[35] A. Ruhe, The rational krylov algorithm for nonsymmetric eigenvalue problems, III: complex shifts for real matrices, BIT 34 (1994) 165–176.

[36] J. Shumway, L.R.C. Fonseca, J.P. Leburton, R.M. Martin, D.M. Ceperley, Electronic structure of self-assembled quantum dots: comparison between density functional theory and diffusion quantum monte carlo, Physica E 8 (2000) 260–268.

[37] G.L.G. Sleijpen, A.G.L. Booten, D.R. Fokkema, H.A. van der Vorst, Jacobi–Davidson type methods for generalized eigenproblems and polynomial eigenproblems, BIT 36 (3) (1996) 595–633.

[38] G.L.G. Sleijpen, H.A. van der Vorst, A Jacobi-Davidson iteration method for linear eigenvalue problems, SIAM J. Matrix Anal. Appl. 17 (2) (1996) 401–425.

[39] G.L.G. Sleijpen, H.A. van der Vorst, M. van Gijzen, Quadratic eigenproblems are no problem, SIAM News 29 (1996) 8–9.

[40] J. Stoer, R. Bulirsch, Introduction to Numerical Analysis, Springer Verlag, Berlin, 2002.

[41] The MathWorks, Inc., Using MATLAB., 2000.

[42] F. Tisseur, Backward error analysis of polynomial eigenvalue problems, Linear Algebra Appl. 309 (2000) 339–361.

[43] W. Wang, T.-M. Hwang, W.-W. Lin, J.-L. Liu, Numerical methods for semiconductor hetrostructures with band nonparabolicity, Journal of Computational Physics 190 (2003) 141–158.

[44] A.J. Williamson, A. Zunger, InAs quantum dots: predicted electronic structure of free-standing versus GaAs-embedded structures, Phys. Rev. B 59 (1999) 15819–15824.